MICROCOPY RESOLUTION TEST CHART

AD-A179 410

Determining the 3-D Motion of a
Rigid Surface Patch without Correspondence,
under Perspective Projection:
I. Planar Surfaces. II. Curved Surfaces.

John (Yiannis) Aloimonos and Isidore Rigoutsos
Department of Computer Science
The University of Rochester
Rochester, NY 14627

TR 178
December 1985

DTIC
ELECTE
APR 2 3 1987
D
D

Department of Computer Science
University of Rochester
Rochester, New York 14627

87 4   22 106

Determining the 3-D Motion of a
Rigid Surface Patch without Correspondence,
under Perspective Projection:
I. Planar Surfaces. II. Curved Surfaces.

John (Yiannis) Aloimonos and Isidore Rigoutsos
Department of Computer Science
The University of Rochester
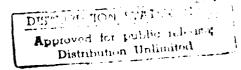Rochester, NY 14627

TR 178
December 1985

Abstract

A method is presented for the recovery of the 3-D motion parameters of
a rigidly moving textured surface. The novelty of the method is based
on the following two facts:

1) no point-to-point correspondences are used; and

2) "stereo" and "motion" are combined in such a way that no
   correspondence between the left and the right stereo pairs is
   required.

Determining the 3-D motion of a rigid surface patch without correspondence, under perspective projection.
I. Planar Surfaces: Theory and Experiments
II. Curved surfaces: Theory

John (Yiannis) Aloimonos and Isidore Rigoutsos
Department of Computer Science
The University of Rochester,
Rochester, New York 14627.

Abstract

`A method is presented for the recovery of the 3D motion parameters of a rigidly moving textured surface. The novelty of the method is based on the following two facts :

1) no point-to-point correspondences are used, and
2) "stereo" and "motion" are combined in such a way that no correspondence between the left and the right stereo pairs is required.

1. Introduction

An important problem in Computer Vision is to recover the 3-D motion of a moving object from its successive images. Dynamic visual information can be produced by a sensor moving through the environment and/or by independently moving objects in the observer's visual field. The interpretation of such dynamic imagery information consists of dynamic segmentation, recovery of the 3-D motion (of the sensor and the objects in the environment) as well as determination of the structure of the environmental world. The results of such an interpretation can be used to control behavior as for example in robotics, tracking, and autonomous navigation. Up to now there have been, basically, three aproaches towards the solution of this problem :

1) The first assumes the dynamic image to be a three-dimensional function of two spatial arguments and a temporal argument. Then if this function is locally well - behaved and its spatiotemporal derivatives are computable, the image velocity or optical flow may be computed [7, 9, 10, 17, 23, 35, 39].

2) The second method for measuring image motion considers the cases where the motion is "large" and the previous technique is not applicable. In these instances the measurement technique relies upon isolating and tracking highlights or feature points in the image through time. In other words operators are applied on both

dynamic frames which output a set of points in both images, and then the correspondence problem between these two sets of points has to be solved (i.e. finding which points on both dynamic frames are due to the projection of the same world point)[3,21a,21b,6,32,33].

In both the above approaches, after the optical flow field or the discrete displacements field (which can be sparse) are computed, then algorithms are constructed for the determination of the three-dimensional motion , based on the optical flow or discrete displacements values [1, 4, 5, 8, 18, 19, 24, 25,26, 27, 28, 29, 30, 32,33, 34, 36, 38].

3) The three-dimensional motion parameters are computed directly from the spatial and temporal derivatives of the image intensity function. In other words, if f is the intensity function and (u,v) the optical flow at a point, then the equation $f_x u + f_y v + f_t = 0$ holds approximately. All the methods in the category are based on the substitution of the optical flow values in terms of the three dimensional motion parameters in the above equation, and there is very good work in this direction [22, 11, 2].

As the problem has been formulated over the years, one camera is used and so the three dimensional motion parameters that have to be computed and can be compute, are five (two for the direction of translation and three for the rotation). In our approach, we consider a binocular observer, and so all six parameters of the motion can be recovered.

2. Motivation and Previous Work

The basic motivation for this research is the fact that optical flow (or discrete displacement) fields produced from real images by existing techniques are corrupted by noise and are partially incorrect [33]. Most of the algorithms in the litterature that use the retinal motion field to recover three-dimensional motion fail when the input (retinal motion) is noisy. Some algorithms work reasonably for images in a specific domain.

Some researchers [26, 40, 41, 42, 8, 43] developed sets of nonlinear equations with the three-dimensional motion parameters as unknowns, which are solved by iterations and initial guessing. These methods are very sensitive to noise, as it is reported in [26, 40, 8, 43]. On the other hand, other researchers [30, 18] developed methods that do not require the solution of nonlinear systems, but the solution of linear ones. Despite that, under the presence of noise, the results are not satisfactory [30, 18].

Bruss and Horn [5] presented a least-squares formalism that tried to compute the motion parameters by minimizing a measure of the difference between the input optic flow and the predicted one from the motion parameters. The method, in the general case, results in solving a system of nonlinear equations with all the inherent difficulties in such a task, and it seems to have good behavior with respect to noise only when the noise in the optical flow field has a particular distribution. Prazdny, Rieger, and Lawton presented methods based on the separation of the optical flow field in its translational and rotational components, under different assumptions [24, 25]. But difficulties are reported with the approach of Prazdny in the present of noise [44], while the methods of Rieger and Lawton require the presence of occluding boundaries in the scene, something which cannot be guaranteed. Finally, Ullman in his pioneering work [32] presented a local analysis, but his approach seems to be sensitive to noise, because of its local nature.

Several other authors [19, 38] use the optical flow field and its first and second spatial derivatives at corresponding points to obtain the motion parameters. But

these derivatives seem to be unreliable with noise, and there is no known algorithm which can determine them reasonably in real images. Others [1] follow an approach based partially on local interpretation of the flow field, but it can be proved [34] that any local interpretation of the flow field is unstable.

At this point it is worth noting that all the aforementioned methods assume an unrestricted motion (translation and rotation). In the case of restricted motion (only translation), a robust algorithm has been reported by Lawton [45], which was successfully applied to some real images. His method is based on a global sampling of an error measure that corresponds to the potential position of the focus of expansion (FOE); finally, a local search is required to determine the exact location of the minimum value. However, the method is time-consuming, and is likely to be very sensitive to small rotations. Also the inherent problems of correspondence, in the sense that there may be drop-ins or drop-outs in the two dynamic frames, is not taken into account. All in all, most of the methods presented up to now for the computation of three-dimensional motion depend on the value of flow or retinal displacements. Probably there is no algorithm until now that can compute retinal motion reasonably (for example, 10% accuracy) in real images.

Even if we had some way, however, to compute retinal motion in a reasonable (acceptable) fashion, i.e., with at most an error of 10%, for example, all the algorithms proposed to date that use retinal motion as input, would still produce non-robust results. The reason for this is the fact that the motion constraint (i.e., the relation between three-dimensional motion and retinal displacements) is very sensitive to small perturbations ([47]). Table 1 shows how the error of motion parameters grows as the error in image point correspondence increases when 8-point correspondence is used, and Table 2 shows the same relationship when 20-point correspondence is used with 2.5% error on point correspondences based on a recent algorithm of great mathematical elegance.

(Tables 1 and 2 are from [30].)

Table 1: Error of motion parameters for 8-point correspondence for 2.5% error in point correspondence.

Error of E (essential parameters)	73.91%
Error of rotation parameters	38.70%
Error of translations	103.60%

Table 2: Error of motion parameters for 20-point correspondence for 2.5% error in point correspondence.

Error of E (essential parameters)	19.49%
Error of rotation parameters	2.40%
Error of translations	29.66%

It is clear from the above tables that the sensitivity of the algorithm in [30] to small errors is very high. It is worth noting at this point that the algorithm in [30] is solving linear equations, but the sensitivity to error in point correspondences is not improved with respect to algorithms that solve non-linear equations. Also, it is worth mentioning at this point that the same behaviour is present in the algotithms that compute 3-D motion in the case of planar surfaces [30].

Finally, the third approach, which computes directly the motion parameters from the spatiotemporal derivatives of the image intensity function, gets rid of the correspondence problem and seems very promising. In [11, 22, 14], the behavior with respect to noise is not discussed. But extensive experiments [31] implementing

the algorithms presented in [2] show that noise in the intensity function affects the computed three-dimensional motion parameters a great deal. We should also mention that the constraint $f_x u + f_y v + f_t = 0$ is a very gross approximation of the actual constraint under perspective projection [46]. So, despite the fact that no correspondences are used in this approach, the resulting algorithms seem to have the same sensitivity to small errors in the input as in the previous cases. This fact should not be surprising, because even if we avoid correspondences, the constraint between three-dimensional motion and retinal motion (regardless of whether the retinal motion is expressed as optic flow or the spatiotemporal variation of the image intensity function) will be essentially the same when one camera is used (monocular observer, traditional approach). This constraint cannot change, since it relates three-dimensional motion to two-dimensional motion through projective geometry.

So, as the problem has been formulated (monocular observer), it seems to have a great deal of difficulty. This is again not surprising, and the same problem is encountered in many other problems in computer vision (shape from shading, structure from motion, stereo, etc.). There has recently been an approach to combine information from different sources in order to achieve uniqueness and robustness of low-level visual computations [47]. With regard to the three-dimensional motion parameters determination problem, why not combine motion information with some other kind of information? It is clear that in this case the constraints won't be the same, and there is some hope for robustness in the computed parameters. As this other kind of information that should be combined with motion, we choose stereo.

The need for combining stereo with motion has recently been appreciated by a number of researchers [13, 37, 12, 47]. Jenkin and Tsotsos, [13], used stereo information for the computation of retinal motion, and they presented good results for their images. Waxman et al. [37] presented a promising method for dynamic stereo, which is based on the comparison of image flow fields obtained from cameras in known relative motion, with passive ranging as goal. Whitman Richards [48] is combining stereo disparity with motion in order to recover correct three-dimensional configurations from two-dimensional images (othography-vergence). Finally, Huang and Blostein [12] presented a method for three-dimensional motion estimation that is based on stereo information. In their work, the static stereo problem as well as the three-dimensional matching problem have to be solved before the motion estimation problem. The emphasis is placed on the error analysis, since the amount of noise (in typical image resolutions) in the input of the motion estimation algorithm is very large.

So a natural question arises: is it possible to recover three-dimensional motion from images without having to go through the very difficult correspondence problem? And if such a thing is possible, how immune to noise will the algorithm be? In this paper, we prove that if we combine stereo and motion in some sense and we avoid any static or dynamic correspondence, then we can compute the three-dimensional motion of a moving object. At this point, it is worth noting recent results by Kanatani [15, 16] that deal with finding the three-dimensional motion of planar contours in small motion, without point correspondences. These methods seem to suffer from numerical errors a great deal, but they have a great mathematical elegance.

As the problem has been formulated over the years, usually one camera is used and so the 3-D motion parameters that can be computed are five : 2 for the direction of translation and 3 for the rotation. In our approach, we assume a binocular observer and so we recover 6 motion parameters : 3 for the translation and 3 for the rotation.

With the traditional one camera approach for the estimation of the 3-D motion parameters of a rigid planar patch, it was just mentioned [26],that one should use the image point correspondences for object points not on a single planar patch when

estimating 3-D motions of rigid objects. But it was not known, how many solutions there were, what was the minimum number of points and views needed to assure uniqueness and how could those solutions be computed without using any iterative search (i.e. without having to solve non-linear systems). It was proved [27,28,30] that there are exactly two solutions for the 3-D motion parameters and plane orientations, given at least 4 image point correspondences in two perspective views, unless the 3x3 matrix containing the canonical coordinates of the second kind [20] for the Lie transformation group that characterizes the retinal motion field of a moving planar patch, has multiple singular values. However, the solutions are unique if three views of the planar patch are given or two views with at least two planar patches. In our approach, the duality problem does not exist for two views, since two cameras are used (and so the analysis is done in 3-D).

In this paper, we present a method for the recovery of the 3-D motion of a rigidly moving surface patch, by a binocular observer without using correspondence neither for the stereo nor for the motion. We first analyze the case of planar surfaces and then we develop the theory for any surface.

The organization of the paper is as follows: the next **Section 3** describes how to recover the structure and depth of a set of 3-D planar points from their images in the left and right flat retinae, without using any point correspondences. We also discuss the effect of noise in the procedure and we describe a method for the improvement of the two camera model using three cameras (trinocular observer).
Section 4 gives a method for the recovery of the 3-D direction of translation of a translating set of planar points from their images without using any correspondence; it furthermore introduces the reader to **Section 5** which deals with the solution of the general problem (the case where the set of 3-D planar points is moving rigidly -- i.e. translating and rotating).
Section 6 describes the theory for the determination of 3-D motion for any kind of surface that moves with an unrestricted motion.
Finally **Section 7** describes experiments as well as the effect of noise in the methods developed for the case of planar surfaces. Experiments for the case of nonplanar surfaces are under development.

3. Stereo without correspondence

In this section we present a method for the recovery of the 3-D parameters for the set of 3-D planar points from their left and right images without using any point-to-point correspondence; instead we consider all point correspondences at once and so there is no need to solve the difficult correspondence problem in the case of the static stereo.

Let an orthogonal cartesian coordinate system OXYZ be fixed with respect to the left camera, with O at the origin (O being also the nodal point of the left eye) and the Z-axis pointing along the optical axis.

Let the image plane of the left camera be perpendicular to the Z-axis at the point $(0,0,f)$, (focal length $= f$).

Let the nodal point of the right camera be at the point $(d,0,0)$ and its image plane be identical to the left one; the optical axis of the right camera (eye) points also along the Z-axis and passes through point $(d,0,0)$ (see **Figure 1.**).

Consider a set of 3-D points $A = \{ (X_i, Y_i, Z_i) / i = 1,2,3 ... n \}$ lying on the same plane(see **Figure 1.**), the latter being described by the equation :

$$Z = p*X + q*Y + c$$

Let O_l, O_r be the origins of the two-dimensional orthogonal coordinate systems on each image plane; these origins are located on the left and right optical axes while the corresponding coordinate systems have their y-axes parallel to the axis OY, and their x-axes parallel to OX.

Finally let $\{(x_{li}, y_{li}) \; / \; i = 1,2,3 \dots n\}$ and $\{(x_{ri}, y_{ri}) \; / \; i = 1,2,3 \dots n\}$ be the projections of the points of set A on the left and right retinae, respectively, i.e.

$$x_{li} = \frac{f*X_i}{Z_i} \quad (1) \qquad\qquad y_{li} = \frac{f*Y_i}{Z_i} \quad (2) \qquad / \; i = 1,2,3 \dots n$$

$$x_{ri} = \frac{f*(X_i - d)}{Z_i} \quad (3) \qquad\qquad y_{ri} = \frac{f*Y_i}{Z_i} \quad (4) \qquad / \; i = 1,2,3 \dots n$$

Let (x_{li}, y_{li}) and (x_{ri}, y_{ri}) be corresponding points in the two frames. Then we have that.

$$x_{li} - x_{ri} = \frac{f*d}{Z_i} \quad (5)$$

$$y_{li} = y_{ri} \quad (6)$$

where Z_i, the depth of the 3-D point having those projections.

In the sequel, we prove that the quantity

$$\sum_{i=1}^{n} \frac{y_{li}^k}{Z_i}$$

is directly computable without using any point correspondence between the left and right frames. We proceed with the following propositions:

3.1 Proposition : Using the aforementioned nomenclature the quantity

$$\sum_{i=1}^{n} \frac{y_{li}^k}{Z_i}$$

where

$$k \geq 0 \;\wedge\; k \neq \frac{m}{2*n} \;, \qquad m,n \in Z - \{0\},$$

is directly computable.

< Proof > We have that

$$\sum_{i=1}^{n} \frac{y_{li}^{k}}{Z_{i}} = (\text{from equation (5)}) = \sum_{i=1}^{n} y_{li}^{k} * \frac{(x_{li} - x_{ri})}{f*d} =$$

$$= \sum_{i=1}^{n} \frac{x_{li}*y_{li}^{k}}{f*d} - \sum_{i=1}^{n} \frac{x_{ri}*y_{li}^{k}}{f*d} = (\text{from equation (6)}) =$$

$$= \sum_{i=1}^{n} \frac{x_{li}*y_{li}^{k}}{f*d} - \sum_{i=1}^{n} \frac{x_{ri}*y_{ri}^{k}}{f*d} .$$

Thus,

$$\sum_{i=1}^{n} \frac{y_{li}^{k}}{Z_{i}} = \sum_{i=1}^{n} \frac{x_{li}*y_{li}^{k}}{f*d} - \sum_{i=1}^{n} \frac{x_{ri}*y_{ri}^{k}}{f*d} \qquad (7)$$

From equation (7) the claim is obvious.

3.2 Proposition : Using the aforementioned nomenclature, the parameters p. q and c of the plane in view are directly computable without using any point-to-point correspondence between the two frames.

<**Proof**> The equation of the world plane when expressed in terms of the coordinates of the left frame, becomes :

$$\frac{1}{Z} = (f - p*x_{l} - q*y_{l}) * \frac{1}{c*f} \qquad (8)$$

So, from equation (8) it follows that :

$$\frac{1}{Z_{i}} = (f - p*x_{li} - q*y_{li}) * \frac{1}{c*f} \qquad / \ i = 1,2,3 \dots n \qquad (9)$$

Now, we have :

$$\sum_{i=1}^{n} \frac{y_{li}^{k}}{Z_{i}} = \sum_{i=1}^{n} (f - p*x_{li} - q*y_{li}) * \frac{y_{li}^{k}}{c*f}$$

or

$$\sum_{i=1}^{n} \frac{y_{li}^{k}}{Z_{i}} = \frac{1}{c} * \sum_{i=1}^{n} y_{li}^{k} - \frac{1}{c*f} * [\sum_{i=1}^{n} p*x_{li}*y_{li}^{k} + \sum_{i=1}^{n} q*y_{li}*y_{li}^{k}] \qquad (10)$$

The left-hand side of equation (10) has been shown to be computable without using any point-to-point correspondence (see Proposition 3.1).

If we write equation (10) for three different values of k, we obtain the following **linear** system in the unknowns p,q,c which in general has a unique solution (except for the case where the projection of all points of set A, have the same y-coordinate in both frames) :

$$\sum_{i=1}^{n} \frac{x_{li} * y_{li}^{k1}}{f * d} - \sum_{i=1}^{n} \frac{x_{ri} * y_{ri}^{k1}}{f * d} = \frac{1}{c} * \sum_{i=1}^{n} y_{li}^{k1} - \frac{1}{c*f} * [\sum_{i=1}^{n} p*x_{li}*y_{li}^{k1} + \sum_{i=1}^{n} q*y_{li}*y_{li}^{k1}] \qquad (11)$$

$$\sum_{i=1}^{n} \frac{x_{li} * y_{li}^{k2}}{f * d} - \sum_{i=1}^{n} \frac{x_{ri} * y_{ri}^{k2}}{f * d} = \frac{1}{c} * \sum_{i=1}^{n} y_{li}^{k2} - \frac{1}{c*f} * [\sum_{i=1}^{n} p*x_{li}*y_{li}^{k2} + \sum_{i=1}^{n} q*y_{li}*y_{li}^{k2}] \qquad (12)$$

$$\sum_{i=1}^{n} \frac{x_{li} * y_{li}^{k3}}{f * d} - \sum_{i=1}^{n} \frac{x_{ri} * y_{ri}^{k3}}{f * d} = \frac{1}{c} * \sum_{i=1}^{n} y_{li}^{k3} - \frac{1}{c*f} * [\sum_{i=1}^{n} p*x_{li}*y_{li}^{k3} + \sum_{i=1}^{n} q*y_{li}*y_{li}^{k3}] \qquad (13)$$

where we used equation (7) to the left hand sides.

The solution of the above system recovers the structure and the depth of the points of set A without any correspondence and this is the conclusion of Proposition 3.2.

3.3 Practical Considerations

We have implemented the above method for different values of k_1, k_2, k_3 and especially for the cases:

a)	$k_1 = 0$	$k_2 = 1/3$	$k_3 = 2/3$
b)	$k_1 = 0$	$k_2 = 1/3$	$k_3 = 1/5$

The noiseless cases give extremely accurate results.

Before we proceed, we must explain what we mean by noise introduced in the images. When we say that one frame (left or right) has noise of $a\%$, we mean that if the plane contains N projection points we added $[(N*a)/100]$ randomly distributed points. (*Note:* [] denotes the integer part of its argument).

When the noise in both frames is kept below 2% then the results are still very satisfactory. When the noise exceeds 5% then only the value of p gets corrupted, but the values of q and c remain very satisfactory. To correct this and get satisfactory results for high noise percentages, we devised the following method that uses three cameras :

"We consider the three camera configuration system as in **Figure 2.**, where the top camera has only vertical displacement with respect to the left one. If all three images are corrupted by noise (ranging from 5% to 20%) then application of the algorithm (*Proposition 3.2*) to the left and top frames will give very reasonable values for p and c and corrupt q, which q, as well as c, are accurately computed from the application of the same algorithm to the right and left frames ".

So, by applying our stereo (without correspondence) algorithm to the 3-camera configuration vision system, we obtain accurate results for the parameters describing the 3-D planar patch, even for noise percentages of 20% or slightly more.

and for different amounts of noise in the different frames. **Section 7** describes relevant experiments.

4.Recovering the direction of translation.

Here we treat the case where the points of set A just rigidly translate, and we wish to recover the direction of the translation. In this case, the depth is not needed but the orientation of the plane is required. The general case is treated in the next **Section 5.**

4.1 Technical prerequisites.

Consider a coordinate system OXYZ fixed with respect to the camera; O coincides with the nodal point of the eye, while the image plane is perpendicular to the Z-axis (focal length $=f$), that is pointing along the optical axis (see **Figure 3.**).

Let us represent points on the image plane with small letters (e.g (x,y)) and points in the world with capital ones (e.g. (X,Y,Z)).

Let us consider a point $P = (X_1, Y_1, Z_1)$ in the world, with perspective image (x_1, y_1), where $x_1 = (f * X_1) / Z$ and $y_1 = (f * Y_1) / Z$.

If the point P moves to the position $P' = (X_2, Y_2, Z_2)$ with

$$X_2 = X_1 + \Delta X \qquad\qquad (14)$$
$$Y_2 = Y_1 + \Delta Y \qquad\qquad (15)$$
$$Z_2 = Z_1 + \Delta Z \qquad\qquad (16)$$

then we desire to find the direction of the translation $(\Delta X/\Delta Z, \Delta Y/\Delta Z)$.

If the perspective image of P' is (x_2, y_2), then the observed motion of the world point in the image plane is given by the displacement vector : (x_2-x_1, y_2-y_1) (which in the case of very small motion is also known as "optical flow").

We can easily prove that :

$$x_2 - x_1 = \frac{f * \Delta X - x_1 * \Delta Z}{Z_1 - \Delta Z} \qquad\qquad (17)$$

$$y_2 - y_1 = \frac{f * \Delta Y - y_1 * \Delta Z}{Z_1 - \Delta Z} \qquad\qquad (18)$$

Under the assumption that the motion in depth is small with respect to the depth, the equations above become :

$$x_2 - x_1 = \frac{f * \Delta X - x_1 * \Delta Z}{Z_1} \qquad\qquad (19)$$

$$y_2 - y_1 = \frac{f^* \Delta Y - y_1 {}^* \Delta Z}{Z_1} \qquad (20)$$

The above equations relate the retinal motion (left-hand sides) to the world motion ΔX, ΔY, ΔZ.

4.2 Detecting 3-D direction of translation without correspondence.

Consider again a coordinate system OXYZ fixed with respect to the camera as in **Figure 4.**, and let $A = \{ (X_i, Y_i, Z_i) / i = 1,2,3 \dots n \}$, such that

$$Z_i = p^* X_i + q^* Y_i + c \qquad / i = 1,2,3 \dots n$$

that is the points are planar. Let the points translate rigidly with translation $(\Delta X, \Delta Y, \Delta Z)$, and let $\{ (x_i, y_i) / i = 1,2,3 \dots n \}$ and $\{ (x_i', y_i') / i = 1,2,3, \dots n \}$ be the projections of the set A before and after the translation, respectively.

Consider a point (x_i, y_i) in the first frame which has a corresponding one (x_i', y_i') in the second (dynamic) frame.

For the moment we do not worry about where the point (x_i', y_i') is, but we do know that the following relations hold between these two points :

$$x_i' - x_i = \frac{f^* \Delta X - x_i {}^* \Delta Z}{Z_i} \qquad (21)$$

$$y_i' - y_i = \frac{f^* \Delta Y - y_i {}^* \Delta Z}{Z_i} \qquad (22)$$

where Z_i is the depth of the 3-D point whose projection (on the first dynamic frame) is the point (x_i, y_i). Taking now into account that

$$\frac{1}{Z_i} = \frac{f - p^* x_i - q^* y_i}{c^* f} \qquad (23)$$

the above equations become :

$$x_i' - x_i = (f^* \Delta X - x_i {}^* \Delta Z) {}^* \frac{f - p^* x_i - q^* y_i}{c^* f} \qquad (24)$$

$$y_i' - y_i = (f^* \Delta Y - y_i {}^* \Delta Z) {}^* \frac{f - p^* x_i - q^* y_i}{c^* f} \qquad (25)$$

If we now write equation (24) for all the points in the two dynamic frames and sum the resulting equations up, we take :

$$\sum_{i=1}^{n} (x_i' - x_i) = \sum_{i=1}^{n} [\,(f^* \Delta X - x_i^* \Delta Z\,)^* \frac{f - p^* x_i - q^* y_i}{c^* f}\,]$$

or

$$\sum_{i=1}^{n} (x_i' - x_i) = \sum_{i=1}^{n} [\,\frac{f^* (f - p^* x_i - q^* y_i)^* \Delta X - x_i^* (f - p^* x_i - q^* y_i)^* \Delta Z}{c^* f}\,] \qquad (26)$$

Similarly, if we do the same for equation (25), we take :

$$\sum_{i=1}^{n} (y_i' - y_i) = \sum_{i=1}^{n} [\,(f^* \Delta Y - y_i^* \Delta Z\,)^* \frac{f - p^* x_i - q^* y_i}{c^* f}\,]$$

or

$$\sum_{i=1}^{n} (y_i' - y_i) = \sum_{i=1}^{n} [\,\frac{f^* (f - p^* x_i - q^* y_i)^* \Delta Y - y_i^* (f - p^* x_i - q^* y_i)^* \Delta Z}{c^* f}\,] \qquad (27)$$

At this point it has to be understood that equations (26) and (27) do not require our finding of any correspondence.

By dividing equation (26) by equation (27), we get :

$$\frac{\sum_{i=1}^{n} x_i' - \sum_{i=1}^{n} x_i}{\sum_{i=1}^{n} y_i' - \sum_{i=1}^{n} y_i} = \frac{\sum_{i=1}^{n} [\,\frac{\Delta X}{\Delta Z}^* f^* (f - p^* x_{li} - q^* y_{li}) - (f - p^* x_{li} - q^* y_{li})^* x_{li}\,]}{\sum_{i=1}^{n} [\,\frac{\Delta Y}{\Delta Z}^* f^* (f - p^* x_{li} - q^* y_{li}) - (f - p^* x_{li} - q^* y_{li})^* y_{li}\,]} \qquad (28)$$

Equation (28) is a linear equation in the unknowns $\Delta X / \Delta Z$, $\Delta Y / \Delta Z$ and the coefficients consist of expressions involving summations of point coordinates in both dynamic frames; for the computation of the latter no establishment of any point correspondences is required.

So, if we consider a binocular observer, applying the above procedure in both left and right "eyes", we get two linear equations (of the form of equation (28)) in the two unknowns $\Delta X / \Delta Z$, $\Delta Y / \Delta Z$, which constitute a linear system that in general has a unique solution.

4.2 What the previous method is not about

If one is not carefull when analyzing the previous method, then he might think that all the method does, is to correspond the center of mass of the image points before the motion with the center of mass of the image points after the motion, and then based on that retinal motion to recover three dimensional motion. But this is wrong, because perspective projection does not preserve simple ratios, and so the

center of mass of the image points before the motion does not correspond to the center of mass of the image points after the motion. All the above method does, is aggregation of of the motion constraints; it does not correspond centers of mass.

4.3 Practical considerations.

We have implemented the above method with a variety of planes as well as displacements; noiseless cases give exremely accurate results, while cases with noise percentages up to 20% (even with different amounts of noise in all four frames (first left and right - second left and right)) give very satisfactory results (an error of at most 5%). **Section 7** describes relevant experiments. We now proceed considering the general case.

5. Determining unrestricted 3-D motion of a rigid planar patch without point correspondences.

Consider again the imaging system (binocular) of **Figure 4.**, as well as the set $A = \{ (X_i, Y_i, Z_i) / i = 1,2,3 \dots n \}$ such that :

$$Z_i = p*X_i + q*Y_i + c \qquad / \quad i = 1,2,3 \dots n$$

i.e. the points are planar; let B be the plane on which they lie. Suppose that the points of the set A move rigidly in space (translation plus rotation) and they become members of a set $A' = \{ (X_i', Y_i', Z_i') / i = 1,2,3 \dots n \}$. Since all of the points of set A move rigidly, it follows that the points of set A' are also planar; let B' be the (new) plane on which these points lie.

In other words the set A becomes A' after the rigid motion transformation. We wish to recover the parameters of this transformation . From the projection of sets A and A' on the left and right image planes and using the method described in **Section 3** the sets A and A' can be computed. In other words, we know exactly the positions in 3-D of all the points of the sets A and A' (and this has been found without using any point correspondences - **Section 3**).

So, the problem of recovering the 3-D motion has been transformed to the following :
"Given the set A of planar points in 3D and the set A' of new
planar points, which has been produced by applying to the points
of set A a rigid motion transformation, recover that transformation."

Any rigid body motion can be analyzed to a rotation plus a translation; the rotation axis can be considered as passing through any point in the space, but after this point is chosen, everything else is fixed.

If we consider the rotation axis as passing through the center of mass (CM) of the points of set A, then the vector which has as its two endpoints the centers of mass CM_A and $CM_{A'}$ of sets A and A' respectively, represents the *exact* 3-D translation.

So, for the translation we can write

$$\text{translation} \equiv T = (X,Y,Z) = CM_{A'} - CM_A$$

It remains to recover the rotation matrix.

Let, therefore, n_1 and n_2 be the surface normals of the planes B and B'. Then, the angle θ between n_1 and n_2 , where

$$\cos\theta = \frac{n_1 \cdot n_2}{\|n_1\| * \|n_2\|} \quad , \text{ with } ' \cdot ' \text{ the inner-product operator}$$

represents the rotation around an axis $O_1 O_2$ perpendicular to the plane defined by n_1 and n_2, where

$$O_1 O_2 = \frac{n_1 \times n_2}{\|n_1 \times n_2\|} \quad , \text{ with } ' \times ' \text{ the cross-product operator}$$

From the axis $O_1 O_2$ and the angle θ we develop a rotation matrix R_1. The matrix R_1 does not represent the final rotation matrix since we are still missing the rotation around the surface normal. Indeed, if we apply the rotation matrix R_1 and the translation T to the set A, we will get a set A" of points, which is different than A', because the rotation matrix R_1 does not include the rotation around the surface normal n_2.

So we now have a matching problem : on the plane B' we have two sets of points A' and A" respectively, and we want to recover the angle ϕ by which we must rotate the points of set A" (with respect to the surface normal n_2) in order to coincide with those of set A'.

Suppose that we can find angle ϕ. From ϕ and $n2$ we construct a new rotation matrix R_2. The final rotation matrix R can be expressed in terms of R_1 , R_2 as follows :

$$R = R_1 R_2$$

It therefore remains to explain how we can compute the angle ϕ . For this we need the statistical definition of the mean direction.

Definition 1 .

Consider a set $A = \{ (X_i, Y_i) / i = 1,2,3 \ldots n \}$ of points all of which lie on the same plane. Consider the center of mass, CM, of these points to have coordinates (X_{cm}, Y_{cm}). Let also circle (CM,1) be the circle having its center at (X_{cm}, Y_{cm}) and radius of length equal to 1.Let P_i be the interse-ctions of the vectors CMA_i with the circumference of the circle (CM,1), $i = 1,2,3 \ldots n$. Then the "mean direction" of the points of the set A, is defined to be the vector MD, where

$$MD \equiv \sum_{j=1}^{n} CMP_j$$

It is clear that the vector of the mean direction is intrinsically connected with the set of points considered each time, and if the set of points is rotated around an axis perpendicular to the plane and passing through CM, by an angle ω, the new mean direction vector is the previous one rotated by the same angle ω.

So, returning to the analysis of our approach, the angle φ is the angle betwe-en the vectors of mean directions of the sets A' and A" (which have obviously, com-mon CM's).

Moreover, it is obvious that the angle φ, and therefore the rotation matrix R_2, cannot be computed in the case the mean direction is 0 (i.e. in the case the set of points is characterized by a point symmetry).

6. Determining unrestricted 3-D motion of a rigid surface without point correspondences

In this section we consider the problem of the recovery of unrestricted 3-D motion of non-planar surfaces. Again, we consider a set of rigidly moving points, and we assume that the depth information is available. In another work [49], we describe how to recover the depth of a set of non-planar points from their stereo images without having to go through the correspondence problem. So consider the imaging system (binocular) of **Fig. 5** , and a set $A = \{ P_i = (X_i, Y_i, Z_i) / i = 1,2,3 \ldots n \}$ of 3-D non-planar points . The coordinates are with respect to a fixed coordinate system that will be used throughout the paper (we can consider as this system either the system of the left or right camera, or the head frame coordinate system). Applying the method described in [49] , from the left and right images of the points of set A, we can recover the members of A themselves, i.e. their 3-D coordinates. Suppose now that the points of the set A move rigidly in space (translation plus rotation) and that they become members of the set $A' = \{ P'_i = (X'_i, Y'_i, Z'_i) / i = 1,2,3 \ldots n \}$. It is evident that the set A' can be recovered exactly as the set A with the method described in [49] . In other words, the set A becomes A' after the rigid motion transformation. We wish to recover the parameters of this transformation. We have already stated that from the projection of the sets A and A' on the left and right image planes and using the method described in [49] , the sets A and A' can be computed. Hence we know exactly the positions of the points of the sets A and A' (and we came up with this result whithout relying to any point-to-point correspondence). So, for the purposes of this section we will assume that the depth information is available.

From the above discussion , we see that the problem of recovering the 3-D motion has been transformed to the following:

" Given the set A of nonplanar points and the set A ' corresponding to the new positions of the initial points after they have experienced a rigid motion transformation, recover that transformation, without any point-to-point correspondences! "

Any rigid motion can be analyzed to a rotation plus a translation; the rotation axis can be considered as passing through the any point in space, but after this point is chosen, everything else is fixed.

If we consider the rotation axis as passing through the origin of the coordinate system, then if the point (X_i, Y_i, Z_i) ∈ A moves to a new position (X'_i, Y'_i, Z'_i) ∈ A', the following relation holds:

$$(X'_i, Y'_i, Z'_i)^t = R(X_i, Y_i, Z_i)^t + T \quad / i = 1,2,3 \ldots n \quad (29)$$

where R is the 3x3 rotation matrix and $T \equiv (\Delta X, \Delta X, \Delta Z)^t$ is the translation vector. We wish to recover the parameters R and T, without using any point-to-point correspondences.

Let,
$$(X_i, Y_i, Z_i)^t \equiv P_i \text{ and } (X'_i, Y'_i, Z'_i)^t \equiv P'_i \quad / i = 1,2,3 \dots n$$

Then, equation (29) becomes :
$$P_i = RP'_i + T \quad / i = 1,2,3 \dots n$$

Summing up the above n equations and dividing by the total number of points, n, we get :

$$\frac{\sum_{i=1}^{n} P_i}{n} = R \frac{\sum_{i=1}^{n} P_i}{n} + T \quad (30)$$

From equation (30) it is clear that if the rotation matrix R is known, then the translation vector T can be computed. So, in the sequel, we will describe how to recover the rotation matrix R. In order to get rid of the translational part of the motion we shall transform the 3-D points to " free " vectors by subtracting the center-of-mass vector.

Let, therefore, CM_A and $CM_{A'}$ be the center-of-mass vectors of the sets of points A and A' respectively; i.e. $CM_A = \Sigma (P_i / n)$ and $CM_{A'} = \Sigma (P'_i / n)$. We furthermore define:

$$v_i = P_i - CM_A \quad / i = 1,2,3 \dots n$$

$$v'_i = P'_i - CM_{A'} \quad / i = 1,2,3 \dots n$$

With these definitions, the motion equation (29), becomes :

$$v'_i = R v_i \quad / i = 1,2,3 \dots n$$

where R is the (**orthogonal**) rotation matrix.
If we know the correspondences of some points (at least three) then the matrix R can in principle be recovered, and such efforts have been published [12] . But we would like to recover matrix R without using any point correspondences.

Let,
$$v_i = (v_{x_i}, v_{y_i}, v_{z_i}) \quad / i = 1,2,3 \dots n$$
$$v'_i = (v'_{x_i}, v'_{y_i}, v'_{z_i}) \quad / i = 1,2,3 \dots n$$

Note that v_i and v'_i are the position vectors of the members of sets A and A' respectively with respect to their center-of-mass coordinate systems.
We wish to find a quantity that will uniquely characterize the whole sets A and A' in terms of their " relationship " (rigid motion transformation). We have found that the matrix consisting of the second order moments of the vectors v_i and v'_i has these properties. In particular, let

$$V \equiv \begin{bmatrix} \sum_{i=1}^{n} v_{x_i}^2 & \sum_{i=1}^{n} v_{x_i} v_{y_i} & \sum_{i=1}^{n} v_{x_i} v_{z_i} \\[1.5em] \sum_{i=1}^{n} v_{y_i} v_{x_i} & \sum_{i=1}^{n} v_{y_i}^2 & \sum_{i=1}^{n} v_{y_i} v_{z_i} \\[1.5em] \sum_{i=1}^{n} v_{x_i} v_{z_i} & \sum_{i=1}^{n} v_{y_i} v_{z_i} & \sum_{i=1}^{n} v_{z_i}^2 \end{bmatrix}$$

$$V'' \equiv \begin{bmatrix} \sum_{i=1}^{n} v'^2_{x_i} & \sum_{i=1}^{n} v'_{x_i} v'_{y_i} & \sum_{i=1}^{n} v'_{x_i} v'_{z_i} \\[1.5em] \sum_{i=1}^{n} v'_{y_i} v'_{x_i} & \sum_{i=1}^{n} v'^2_{y_i} & \sum_{i=1}^{n} v'_{y_i} v'_{z_i} \\[1.5em] \sum_{i=1}^{n} v'_{x_i} v'_{z_i} & \sum_{i=1}^{n} v'_{y_i} v'_{z_i} & \sum_{i=1}^{n} v'^2_{z_i} \end{bmatrix}$$

From these relations, we have that :

$$V' = \sum_{i=1}^{n} (v'_{x_i}, v'_{y_i}, v'_{z_i})^t \, (v'_{x_i}, v'_{y_i}, v'_{z_i}) =$$

$$= \sum_{i=1}^{n} R(v_{x_i}, v_{y_i}, v_{z_i})^t (v_{x_i}, v_{y_i}, v_{z_i}) \; R^t =$$

$$= R \, V \, R^t$$

So, $$V' = R \, V \, R^t \qquad (31)$$

At this point it should be mentioned that equation (31) represents an invariance between the two sets of 3-D points A and A', since the matrices V and V' are similar. In other words we have discovered that matrix V remains invariant under rigid motion transformation. The reason that the quantity (matrix) V remains invariant is much deeper and very intuitive, and it comes from the principles of Classical Mechanics. Unfortunately, due to lack of space, we are not able to explain at this point how we were led to the discovery of matrix V. The interested reader can consult the Appendix where it is shown how matrix V can be formed from the matrix

corresponding the the second rank moment of inertia tensor. From now on, the recovery of the rotation matrix R is simple and comes from basic Linear Algebra. Furthermore equation (3) implies that the matrices V and V" have the same set of eigenvalues [50].

But since V and V" are symmetric matrices, they can be expanded in their eigenvalue decomposition, i.e. there exist matrices S, T, such that:

$$V = S D S^t \quad (32)$$
$$V" = T D T^t \quad (33)$$

where S, T are orthogonal matrices having as columns the eigenvectors of the matrices V and V" respectively (e.g. i-th column corresponding to the i-th eigenvalue) and D diagonal matrix consisting of the eigenvalues of the matrices V and V". We have to mention at this point that in order to make the decomposition unique we require that the eigenvectors in the columns of matrices S and T be orthonormal.

From equations (31), (32), (33) we derive that matrices T and R S both consist of the orthonormal eigenvectors of matrix V". In other words, the columns of matrices R S and T must be the same, with a possible change of sign. So, the matrix RS is equal to one of eight possible matrices, T_i , i = 1,..,8. Thus, $R = T_i S$ i = 1,...,8. But the rotation matrix is orthogonal and it has determinant equal to one. Furthermore, if we apply matrix R to the set of vectors v_i then we should get the set of vectors v_i '. So, given the above three conditions and Chasles theorem, the matrix R can be computed uniquely.

There is something to be said about the uniqueness properties of the algorithm. When all the eigenvalues of the matrix V have multiplicity one then the problem has a unique solution. When there are eigenvalues with multiplicity more than one, then there is some inherent symmetry in the problem that exhibits some degeneracy properties. For example, if the surface in view (i.e. the surface on which the points lie) is a solid of revolution, then there is an eigenvalue (of the matrix V) with multiplicity 2, and only the eigenvector corresponding to the axis of revolution can be found. The other two eigenvectors define a plane vertical to the axis of revolution. So, in this case there is an inherent degeneracy. We are currently working towards a complete mathematical characterization of the degenerate cases of the problem. We are also developing experiments to test the robustness of the method as well as setting up the equipment for experimentation in natural images.

7. Experiments.

We will describe experiments for both the detection of structure and depth without correspondence and the detection of 3-D motion without correspondence for the case of planar surfaces. Experiments for the case of curved (general) surfaces are under development.

In our experiments, we considered a set of three dimensional planar points, which we projected perspectively in both the left and right frames. From the projections we recover the structure and depth of the 3-D plane using the alogrithm described in Section 3 , or using the projections in three frames . It is clear, that the equations that are used to develop the linear system described in **Section 3**, are based on the assumption that the number of points on (left and right frames), is the same. But in noisy situations, this is not the case. In particular, in real images

operators have first to be applied on all four frames (two before the motion and two after the motion) that will produce points of interest, ([3,6,17,21]) and then the theory developed in this paper is applied to these points.

But any method that will produce points of interest from intensity images is bound to have errors due to the noise in the images and the unpredictable behavior of the intensity function in natural scenes. When we say that the methods that find interesting points in intensity images are bound to errors, we mean that there will be points in the left frame whose corresponding ones have not been found in the right stereo frame, and also there will be points in the first dynamic frame whose corresponding ones have not been found in the second dynamic frame, and vice-versa. So, the number of points will not be the same in the different images. Because of that, our method is bound to have an error, since it is based on the assumption that the number of points is everywhere the same. To reduce this error we do the following: Equations (11), (12), (13) are not affected if both sides are divided by the number of points in all the frames (under the assumption that the number of points is the same in all frames). If now the numbers of points in the left and right fram e are different, say n_{left} and n_{right}, in the static stereo case, then we divide the summations resulting from each of the frames, by the number of points of the corresponding frame, and the resulting equations are (for the static stereo case):

$$\sum_{i=1}^{n} \frac{x_{li}*y_{li}^{k1}}{f*d*n_{left}} - \sum_{i=1}^{n} \frac{x_{ri}*y_{ri}^{k1}}{f*d*n_{right}} = \frac{1}{c*n_{left}} * \sum_{i=1}^{n} y_{li}^{k1} - \frac{1}{c*f*n_{left}} * [\sum_{i=1}^{n} p*x_{li}*y_{li}^{k1} + \sum_{i=1}^{n} q*y_{li}*y_{li}^{k1}] \quad (29)$$

$$\sum_{i=1}^{n} \frac{x_{li}*y_{li}^{k2}}{f*d*n_{left}} - \sum_{i=1}^{n} \frac{x_{ri}*y_{ri}^{k2}}{f*d*n_{right}} = \frac{1}{c*n_{left}} * \sum_{i=1}^{n} y_{li}^{k2} - \frac{1}{c*f*n_{left}} * [\sum_{i=1}^{n} p*x_{li}*y_{li}^{k2} + \sum_{i=1}^{n} q*y_{li}*y_{li}^{k2}] \quad (30)$$

$$\sum_{i=1}^{n} \frac{x_{li}*y_{li}^{k3}}{f*d*n_{left}} - \sum_{i=1}^{n} \frac{x_{ri}*y_{ri}^{k3}}{f*d*n_{right}} = \frac{1}{c*n_{left}} * \sum_{i=1}^{n} y_{li}^{k3} - \frac{1}{c*f*n_{left}} * [\sum_{i=1}^{n} p*x_{li}*y_{li}^{k3} + \sum_{i=1}^{n} q*y_{li}*y_{li}^{k3}] \quad (31)$$

where n_{left} and n_{right} represent the numbers of points in the left and right frames respectively. It is clear that the resulting equations are approximate, but our experiments show that the introduced error is very small. It has to be mentioned , however, that the intrinsic difficulty, appearing in the traditional methods (i.e. stereo, optical flow), of not being able to find corresponding points, exists even in our algorithm but under the form of different numbers of points in the different frames, because of the globality of our approach. However, even considerable differences in the numbers of points among the different frames hardly affects the results. Furthermore, the same technique is applied to the case of motion as well.

Picture 1. shows the projections of a set of planar points on both the left and right frames. The frame on top is the superposition of the left and right frames. The actual parameters of the plane were :
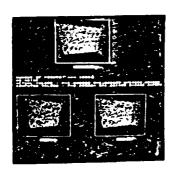p = 0.0 , q = 0.0 , c = 10000, while the number of points was equal to 1000.
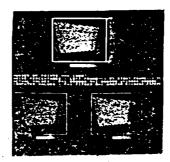We did not include any noise to our pictures.

The computed ones were : $P = -0.0$, $Q = -0.0$, $C = 10000.0$

Picture 1.

Picture 2. shows the projections of a set of planar points on both the left and right frames. The frame on top is the superposition of the left and right frames. The actual parameters of the plane were :
$p = 1.0$, $q = 1.0$, $c = 10000$. while the number of points was equal to 1000.
We did not include any noise to our pictures.
The computed ones were: $P = 0.98$, $Q = 1.00$, $C = 9809.8$

Picture 3. shows the projections of a set of planar points on both the left and right frames. The frame on top is the superposition of the left and right frames. The actual parameters of the plane were :
$p = 1.0$, $q = 1.0$, $c = 10000$. while the number of points was equal to 1000.
We included 5% noise to the left frame and 7% to the right one.
The computed ones were: $P = 1.7$, $Q = 1.2$, $C = 10266.7$

Picture 2.

Picture 3.

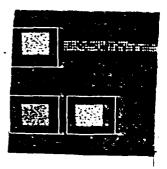**Pictures 4a., 4b.** show the results from the *3-eye method*. Here the projections of a set of 3-D planar points on all the three frames are considered. The actual parameters were:
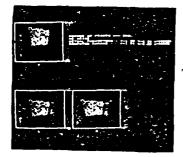$p = 0.0$, $q = 0.0$, $c = 10000$ (**Picture 4a.**) and $p = 1.50$, $q = 2.30$, $c = 10000$
(**Picture 4b.**) respectively. The number of points was equal to 1000. in both pictures.
Picture 4b. did not have any noise, whereas **Picture 4a.** had 5% noise in the left frame and 7% noise in the right and top frames.
The computed ones were: $P = 0.10$, $Q = 0.05$, $C = 10197.0$ and
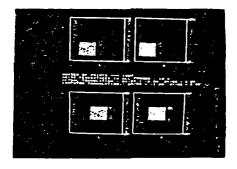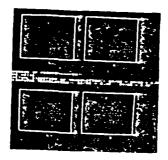$P = 1.51$, $Q = 2.22$, $C = 10000.0$ respectively.
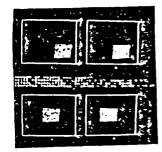
Picture 4a.

Picture 4b.

Pictures 5..6..7..8..9.. show the 3-D motion determination results. In Picture 5.. the two frames at the bottom represent the projections of a set of 3-D planar points on the left and right eyes respectively. The two frames at the top. represent the projections of the same set of points. after it has been translated. The actual direction of translation was equal to (-2.0 , 2.0), and the computed one was (-1.9, 2.0). The noise percentage was equal to 10% in all four frames while the number of points was equal to 1000. At this point it has to be mentioned that the parameters p.q were also computed computed.since the latter are used in the determination of the direction of translation (see also **Section 4**). **Pictures 6.,7.,** represent similar experiments.

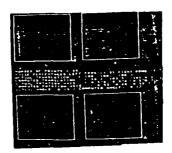
Picture 5.

Picture 6.

Picture 7.

Pictures 8. and **9.** , show experiments determining the general motion . The results were computed according to the method presented in **Section 5.**, and the results were recalculated with respect to the left-camera coordinate system.

Picture 8.

Picture 9.

NOTE: All the parameters involved in the above experiments that have a dimension of length $L^1 M^0 T^0$ are calculated in pixels , where 1 pixel = 100μm.

7. Conclusion and future work.

We have presented a method on how a binocular (or trinocular) observer can recover the structure. depth, and 3-D motion of rigidly moving surface patch without using any static or dynamic point correspondences. We are currently setting up the the experiment for the application of the method in natural images. We are also working towards the development of experiments that will test the robustness of the method presented in section 6 for the recovery of 3-D motion, without point correspondences, in the case of non-planar surfaces.
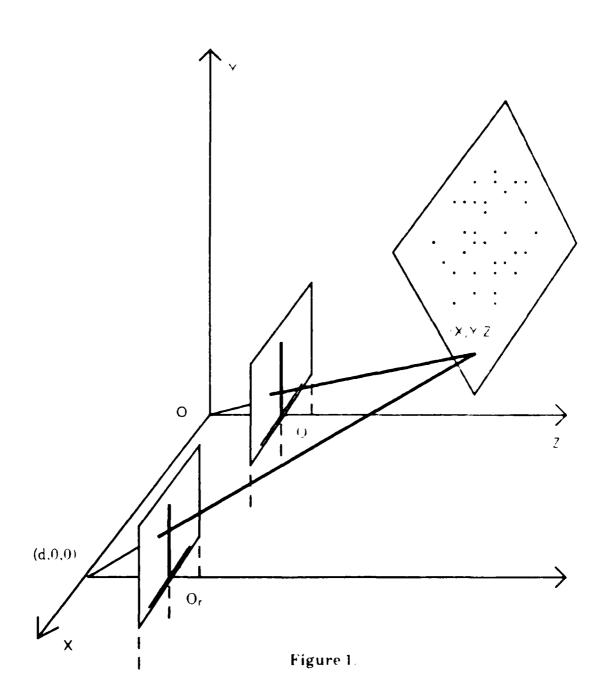
REFERENCES
1. G. Adiv, Determining 3-D Motion and Structure from Optical Flow Generated

from Several Moving Objects. COINS-Tech. Rep. 84-07, July 1984.

2. J. Aloimonos, and C. M. Brown, Direct Processing of Curvilnear Sensor Motion from a Sequence of Perspective Images, Proc. of the IEEE Workshop on Computer Vision: Representation and Control, Annapolis, Maryland,1984.

3. A. Bandyopadhyay, A Multiple Channel Model for the Perception of Optical Flow, IEEE Proc. of the Workshop on Computer Vision Representation and Control, 1984, pp78-82.

4. A. Bandyopadhyay and J. Aloimonos, Perception of Rigid Motion from Spatio-Temporal Derivatives of Optical Flow, Tech. Report 157, Dept. of Computer Science, University of Rochester, 1985.

5. A. Bruss and B.K.P. Horn, Passive Navigation, CVGIP 21, 1983, pp3-20.

6. S.T. Barnard and W.B. Thompson, Disparity analysis of images, IEEE Trans. PAMI, Vol. 2, 1980, pp 333-340.

7. L.S. Davis, Z. Wu and H. Sun, Contour Based Motion Estimation, CVGIP 23, 1983, pp313-326.

8. J.Q.. Fang and T.S. Huang, Solving Three Dimension Small-Rotation Motion Equations: Uniqueness, Algorithms and Numerical Results. CVGIP 26, 1984, pp183-206.

9. R.M. Haralick and J.S.Lee, The Facet Approach to Optical Flow, Proc. Image Understanding Workshop, Arlington, Virginia, June, 1983, pp84-93.

10. B.K.P. Horn and B.G. Schunck, Determining Optical Flow, Artificial Intelligence 17, 1981, pp185-204.

11. T.S. Huang, Three Dimensional Motion Analysis by Direct Matching. Topical Meeting on Machine Vision, Incline Village, Nevada, 1985, ppFA1.1 - FA1.4.

12. T.S. Huang and S.D. Blonstein, Robust Algorithms for Motion Estimation Based on Two Sequential Stereo Image Pairs, Proc, CVPR 1985, pp518-523.

13. M. Jenkin and J. Tsotsos, Applying Temporal Constraints to the Dynamic Stereo Problem, Department of Computer Science, University of Toronto, CVGIP (to be published).

14. T. Kanade, Camera Motion from Image Differentials, Proc. Annual Meeting, Optical Society of America, Lake Tahoe, March 1985.

15. K.-I. Kanatani, Detecting the Motion of a Planar Surface by Line and Surface Integrals, CVGIP, Vol. 29 , 1985, pp13-22.

16. K.-I. Kanatani, Tracing Planar Surface Motion from a Projection without Knowing the Correspondence, CVGIP, Vol. 29, 1985, pp1-12.

17. L, Kitchen and A. Rosenfeld, Gray-Level Corner Detection, TR 887, Computer Science Dept., U. of Maryland, 1980.

18. H.C. Longuet-Higgins, A Computer Algorithm for Reconstructing a Scene from 2 Projections, Nature 293, 1981.

19. H.C.Longuet-Higgins and K.Prazdny, The Interpretation of a Moving Retinal Image, Proc. Royal Society London B 208, 1980, pp385-397.

20. Y. Matsushima, Differentiable Manifolds, New York: Marcel Dekker, 1972.

21. H.P.Moravec, Towards Automatic Visual Obstacle Avoidance, Proc., 5th IJCAI, 1977.

21. H. Nagel, Displacement Vectors Derived from Second Order Intensity Variations in Image Sequences, CVGIP, Vol. 21, 1983,pp85-117.

22. S. Negahdaripour and B.K Horn, Detremining 3-D Motion of Planar Objects from Image Brightness Patterns, IJCAI85, 1985, pp898-901.

23. J.M. Prager and M.A. Arbib, Computing the Optical Flow : The MATCH Algorithm and Prediction, CVGIP, Vol. 21, 1983, pp272-304.

24. K.Prazdny, Determining the Instantaneous Direction of Motion from Optical Flow Generated by Curvilinearly Moving Observer, CVGIP, Vol. 17, 1981, pp 94 97.

25. J.H.Rieger and D.T Lawton, Determining the Instantaneous Axis of Translation from Optic Flow Generated by Arbitrary Sensor Motion, COINS Tech. Report 83-1, January 1983.

26. J.W. Roach and J.K. Aggarwal, Determining the Movement of Objects from a Sequence of Images, PAMI, Vol. 2, No. 6, November 1980, pp554-562.

27. R.Y.Tsai, T.S.Huang and W. Zhu, Estimating Three Dimensional Motion Parameters of a Rigid Planar Patch II: Singular Value Decomposition, IEEE Trans. A.S.S.P. ASSP-30, August 1982, pp525-534.

28. R.Y.Tsai, T.S.Huang, Estimating Three Dimensional Motion Parameters of a Rigid Planar Patch III: Finite Point Correspondences and The Three View Problem, Proc. IEEE Conf. ASSP Paris, May 1982.

29. R.Y.Tsai, T.S.Huang, Uniqueness and Estimation of Three Dimensional Motion Parameters of Rigid Objects with Curved Surfaces, IEEE, Trans. PAMI 6, January 1984, pp13-27.

30. R.Y.Tsai, T.S.Huang, Uniqueness and Estimation of Three Dimensional Motion Parameters of Rigid Objects, Image Understanding 1984, eds Shimmon Ullman and Whitman Richards, Ablex Publ. Co., New Jersey, 1984, pp135.

31. B. Sarachan, Experiments in Rotational Egomotion Calculation, TR 152, Computer Science Dept., U. of Rochester, 1985.

32. S. Ullman, The Interpretation of Visual Motion, Ph.D. Thesis, 1977.

33. S. Ullman, Analysis of Visual Motion by Biological and Computer Systems, IEEE Computer, 14 (8), 1981, pp57-69.

34. S. Ullman, Computational Studies in the Interpretation of Structure and Motion: Summary and Extension, AI Memo No 706, Mit AI Lab., March 1983.

35. S. Ullman and E. Hildreth, The Measurement of Visual Motion, in Physical and Biological Processing of Images (Proc. Int. Symp. Rank Prize Funds London), O.J. Braddick and A.C. Sleigh (ed.), Springer-Verlag, September 1982, pp154-176.

36. B. Yen and T.S. Huang, Determining 3-D Motion and Structure of Rigid Objects Containing Lines Using the Spherical Projection, in Image Sequence Processing & Dynamic Scene Analysis, T.S. Huang (ed.), 1983.

37. A.M. Waxman and S.S. Sinha, Dynamic Stereo: Passive Ranging to Moving Objects from Relative Image Flows, CAR-Tech. Report-74, College Park, Maryland 20742, July 1984.

38. A. Waxman and S. Ullman, Surface Structure and 3-D Motion from Image Flow: A Kinematic Approach, CAR-TR-24, Center for Automation Research, Univ. of Maryland, 1983.

39. K. Wohn and A.Waxman, Contour Evolution, Neighbourhood Deformation and Local Image Flow: Curved Surfaces in MOtion, CAR-TR-134, Computer Vision Lab., Univ. of Massachusetts.

40. Prazdny, K., "Egomotion and relative depth map from optical flow," *Biol. Cybernetics 36, 1980*

41. Nagel, H., "On the derivation of three-dimensional rigid point configurations from image sequences," *Proc., PRIP*, Dallas, TX, 1981.

42. Nagel, H.-H. and B. Neumann, "On three-dimensional reconstruction from and two perspective views," *Proc., 7th IJCAI*, Vancouver, Canada, 1981.

43. Fang, J.Q. and T.S. Huang, "Estimating three-dimensional movement of rigid objects: experimental results," *Proc., 8th IJCAI*, Karlsruhe, West Germany, 1983

44. Jerian, C. and R. Jain, "Determining motion parameters for scenes with translation and rotation," *Proc., Workshop on Motion*, Toronto, Canada, 1983

45. Lawton, D., "Motion analysis via local translational processing," *Proc., Workshop in Computer Vision*, Rindge, NH, 1982.

46. Schunck, B., personal communication, 1984.

47. Aloimonos, J., "Low-level visual computations," Ph.D. thesis, Computer Science Dept., U. Rochester, in preparation.

48. Richards, W., "Structure from stereo and motion," JOSA A 2, 2, February 1985.

49. Aloimonos, J. and Rigoutsos,I.,"Stereo without stereo", forthcoming TR, Dept. of Computer Science, University of Rochester.

50. Stewart,G.,"Introduction toMatrix Computations", Academic Press, 1983.

51. Synge,L.J., and Schild, A.,"Tensor Calculus", Toronto, University of Toronto Press, 1949.

52. Goldstein, H., "Classical Mechanics", Addison -Wesley, 1980.

Figure 1.
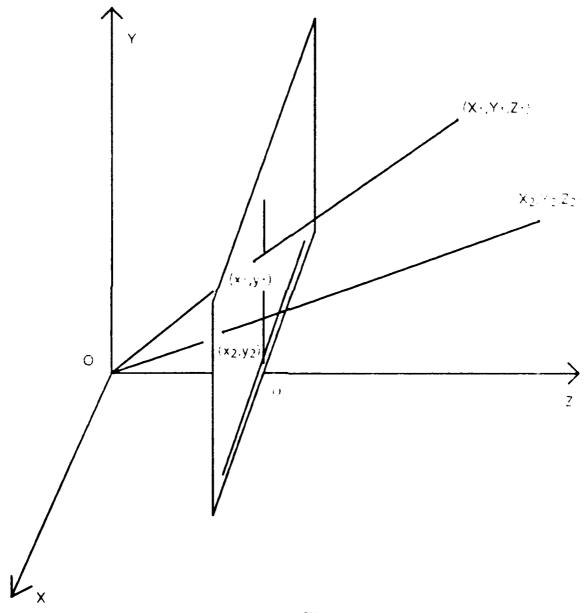
Figure 2.

Figure 3.

Figure 4.
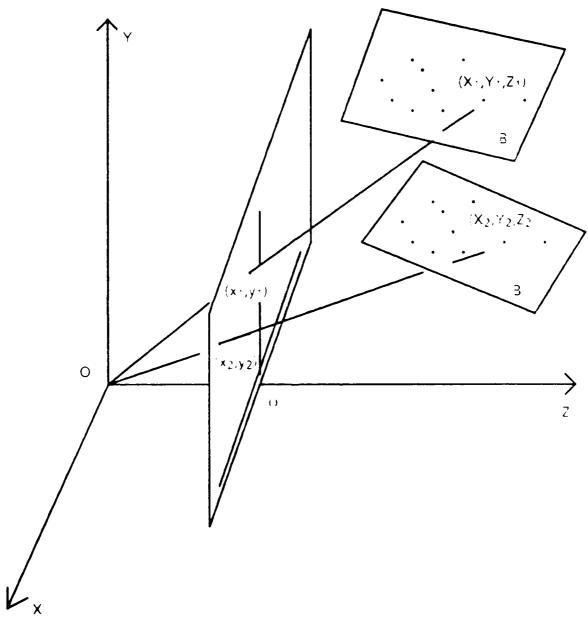
time $t_2 >$ time t_1

Figure 5.

APPENDIX

In order to find this invariant quantity, let us first consider the following:
We know that the quotient of two quantities is often not a member of the same class as the dividing factor, but it may belong to a more complicated class. To support this statement we need only recall that the quotient of two integers is in general a **rational** number. Similarly the quotient of two vectors cannot be defined consistently whithin the calss of vectors; we need a class that is a superset of that of vectors, namely the class of **tensors**. The quantity that is known as moment of inertia of a rigid body with respect to its axis of rotation is defined as:

$$I = \frac{L}{\omega}$$

where I, L, and ω are the moment of inertia of the considered body, the (total) angular momentum of the body and its angular velocity with respect to its axis of rotation, say OO', respectively. It is not therefore surprising to find that I is a *new* quantity, namely **a tensor of the second rank**.

In a Cartesian space of three dimensions, a tensor T of the k-th rank may be defined for our purposes as a quantity having 3^k components $T_{i\,i\,i\quad i}$ that transform under an orthogonal transformation of coordinates, **A** , according to the following relation (see [51]) :

$$T'_{i'_1 i'_2 i'_3 \cdots i'_k}(\mathbf{x}') = \sum_{i_1, i_2 \cdots i_k}^{3} a_{i'_1 i_1} a_{i'_2 i_2} \cdots a_{i'_k i_k} T_{i_1 i_2 i_3 \cdots i_k}(\mathbf{x})$$

By this definition, the $3^2 = 9$ components of a tensor of the second rank transform according to the equation:

$$T_{ij} = \sum_{k,l=1}^{3} a_{ik} a_{jl} T_{kl}$$

If one wants to be rigorous, one **must** distinguish between a second order tensor T and the square matrix formed from its components. A tensor is only defined in terms of its transformation properties under orthogonal coordinate transformations. However, in the case of matrices there is no restriction in the kind of transformations it may experience. But considering the restricted domain of orthogonal transformations, there is a practical as well as important identity. The tensor components and the matrix elements are manipulated in exactly the same fashion; as a matter of fact **for every tensor equation there will be a corresponding matrix equation, and vice versa**. Consider now an orthogonal transformation of coordinates defined by a matrix **A**. Then the components of a square matrix **V** will now be:

$$\mathbf{V}' = \mathbf{A}\,\mathbf{V}\,\mathbf{A}^{t}$$

or equivalently: $V_{ij} = \overset{3}{\Sigma} a_{ik} u_{kl} a_{jl}$

If we now denote by I_{ij} the 3x3 matrix that corresponds to the inertia tensor of the second rank, I, we are able to write the following equation :

$$I' = A\, I\, A^t$$

where,   In the above matrix, m_i is the mass of the i-th " particle " (point) and (x_i,

$$I \equiv \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} \equiv \begin{bmatrix} \Sigma m_i(y_i^2 + z_i^2) & -\Sigma m_i x_i y_i & -\Sigma m_i x_i z_i \\ -\Sigma m_i y_i x_i & -\Sigma m_i(y_i^2 + z_i^2) & -\Sigma m_i y_i z_i \\ -\Sigma m_i z_i x_i & -\Sigma m_i y_i z_i & -\Sigma m_i(y_i^2 + z_i^2) \end{bmatrix}$$

y_i, z_i) $\equiv r_i$ is its position vector with respect to the considered coordinate system.

Restricting ourselves in the center-of-mass coordinate system, with respect to which the rigid motion is viewed as consisting only of a rotational part (see previous discussion and [52]), and recalling that the rotation matrix R defines an orthogonal transformation of the coordinates, we can write:

$$\begin{bmatrix} I'_{xx} & I'_{xy} & I'_{xz} \\ I'_{yx} & I'_{yy} & I'_{yz} \\ I'_{zx} & I'_{zy} & I'_{zz} \end{bmatrix} = R \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} R^t \qquad (1)$$

where the primed and the unprimed factors refer to quantities measured with respect to the center-of-mass coordinate system after and before the transformation (rigid motion) respectively.

Consider now the diagonal matrix:

$$D \equiv \begin{bmatrix} Q & 0 & 0 \\ 0 & Q & 0 \\ 0 & 0 & Q \end{bmatrix}$$, where Q is an arbitrary scalar.

From basic Linear Algebra, it follows that:

$$D = R D R^t \quad (2)$$

The above relation (2) will clearly hold for the case of $Q \equiv \Sigma m_i(x_i^2 + y_i^2 + z_i^2) = \Sigma m_i (r_i \cdot r_i)$, where r_i is the position vector of the i-th " particle " (point) with mass m_i with respect to the center-of-mass coordinate system. At this point recall that the orthogonal transformations preserve **inner** products. Hence, if r_i' is the new position vector with respect to the same coordinate system (center-of-mass), of the i-th "particle" (point), the following equation will obviously hold:

$$r'_i \cdot r'_i = r_i \cdot r_i \quad / i = 1,2,3 \dots n$$

Therefore:

$$Q' \equiv \Sigma m_i (x'_i^2 + y'_i^2 + z'_i^2) = \Sigma m_i(x_i^2 + y_i^2 + z_i^2) \equiv Q$$

and the equation (2) can now be written as follows:

$$D' = R D R^t \quad (3)$$

Note: Recall that the primed quantities refer to the center-of-mass coordinate system after the the rigid motion.

Finally, subtracting equation (3) from equation (1) and recalling from Linear Algebra that :

$$R A_1 R^t - R A_2 R^t = R(A_1 - A_2) R^t$$

for any two matrices A_1, and A_2 of appropriate order, we conclude that:

$$
\begin{bmatrix}
\sum_i m_i x'^2_i & \sum_i m_i x'_i y'_i & \sum_i m_i x'_i z'_i \\[2ex]
\sum_i m_i y'_i x'_i & \sum_i m_i y'^2_i & \sum_i m_i y'_i z'_i \\[2ex]
\sum_i m_i z'_i x'_i & \sum_i m_i y'_i z'_i & \sum_i m_i z'^2_i
\end{bmatrix}
= R
\begin{bmatrix}
\sum_i m_i x^2_i & \sum_i m_i x_i y_i & \sum_i m_i x_i z_i \\[2ex]
\sum_i m_i y_i x_i & \sum_i m_i y^2_i & \sum_i m_i y_i z_i \\[2ex]
\sum_i m_i z_i x_i & \sum_i m_i y_i z_i & \sum_i m_i z^2_i
\end{bmatrix}
R^t
$$

in other words the quantity

$$
\begin{bmatrix}
\sum_i m_i x^2_i & \sum_i m_i x_i y_i & \sum_i m_i x_i z_i \\[2ex]
\sum_i m_i y_i x_i & \sum_i m_i y^2_i & \sum_i m_i y_i z_i \\[2ex]
\sum_i m_i z_i x_i & \sum_i m_i y_i z_i & \sum_i m_i z^2_i
\end{bmatrix}
$$

is an **invariant** under orthogonal transformations, and such a transformation is the rigid motion as viewed from the center-of mass coordinate system. Certainly the moment of inertia matrix I can be used instead of the matrix V (recall section 6), but the matrix V is of a simpler form and so it is better to be used for calculations. The moment of inertia matrix I, facilitates a uniqueness analysis of the problem.

END

5-81

DTIC